```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFFFFFFFFFFFFFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEEEEEEEEEEE     DDD         DDD   FFFFFFFFFFFF
EEEEEEEEEEEE     DDD         DDD   FFFFFFFFFFFF
EEEEEEEEEEEE     DDD         DDD   FFFFFFFFFFFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEE              DDD         DDD   FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD      FFF
```

```
_$

Va
--
00
00
00
00
00
00
00
00
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

```
**FILE**ID**EDFCHF                                    C  1
```

```
EEEEEEEEEE  DDDDDDD     FFFFFFFFFF    CCCCCCCC  HH      HH  FFFFFFFFFF
EEEEEEEEEE  DDDDDDD     FFFFFFFFFF    CCCCCCCC  HH      HH  FFFFFFFFFF
EE          DD      DD  FF          CC          HH      HH  FF
EE          DD      DD  FF          CC          HH      HH  FF
EE          DD      DD  FF          CC          HH      HH  FF
EE          DD      DD  FF          CC          HH      HH  FF
EEEEEEEE    DD      DD  FFFFFFFF    CC          HHHHHHHHHH  FFFFFFFF
EEEEEEEE    DD      DD  FFFFFFFF    CC          HHHHHHHHHH  FFFFFFFF
EE          DD      DD  FF          CC          HH      HH  FF
EE          DD      DD  FF          CC          HH      HH  FF
EE          DD      DD  FF          CC          HH      HH  FF                ....
EE          DD      DD  FF          CC          HH      HH  FF                ....
EEEEEEEEEE  DDDDDDD     FF            CCCCCCCC  HH      HH  FF                ....
EEEEEEEEEE  DDDDDDD     FF            CCCCCCCC  HH      HH  FF                ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
ED
VO

05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
05
```

```
        [ IDENT ('V04-000'),
{ ++
****************************************************************************
**                                                                         *
**  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
**  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
**  ALL RIGHTS RESERVED.                                                   *
**                                                                         *
**  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
**  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
**  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
**  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
**  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
**  TRANSFERRED.                                                           *
**                                                                         *
**  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
**  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
**  CORPORATION.                                                           *
**                                                                         *
**  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
**  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
**                                                                         *
**                                                                         *
****************************************************************************





FACILITY:       VAX/VMS EDF (EDIT/FDL) UTILITY

ABSTRACT:       This facility is used to create, modify, and optimize
                FDL specification files.

ENVIRONMENT:    NATIVE/USER MODE

AUTHOR:         Ken F. Henderson Jr.

CREATION DATE:  27-Mar-1981

MODIFIED BY:

                V03-007 RRB0018         Rowland R. Bradley      10 Mar 1984
                        Changes for signaling errors when user is
                        /NOINT.

                V03-006 KFH0006         Ken Henderson            8 Aug 1983
                        Changes for seperate compilation.

                V03-005 KFH0005         Ken Henderson           14 Apr 1983
                        Changed lib$wait(5.0) to (3.0).
                        Added display of "TOKEN" on errors.

                V03-004 KFH0004         Ken Henderson           26 Jan 1983
                        Fixed signal-vector before $PUTMSG
                        calls by subtracting off PC/PSL.
                        Also changed $PUTMSG of "file not found"
                to "new file will be created".

        V03-003 KFH0003         Ken Henderson           20 Jan 1983
                Removed references to DASH.

        V03-002 KFH0002         Ken Henderson           31 March 1982
                Modified RMS_INPUT_COND_HANDLER to fix
                FT2 QAR 968

        V03-001 KFH0001         Ken Henderson           23-Mar-1982
                Modified RMS_INPUT_COND_HANDLER to fix
                FT2 QAR 694

-- }
```

```
ENVIRONMENT ('LIB$:EDFCHF'),

INHERIT (

'SYS$LIBRARY:STARLET',
'SHRLIB$:FDLPARDEF',
'LIB$:EDFSDLMSG',
'LIB$:EDFSTRUCT',
'LIB$:EDFCONST',
'LIB$:EDFTYPE',
'LIB$:EDFVAR',
'LIB$:EDFEXTERN'

)]

MODULE EDFCHF;
```

```
{ ++

CTRLZ_COND_HANDLER -- Handle user typing control/Z.

This routine checks for control/Z signal from sys$input_cond_handler
and unwinds to the top level if found.
It also is the outermost handler and does a putmsg if it wasn't a ^Z.

CALLING SEQUENCE:

LIB$SIGNAL;

INPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

LIB$MATCH_COND
SYS$UNWIND

ROUTINE VALUE:

SS$_RESIGNAL if unable to handle error.  N/A if able (ignored on unwind).

SIGNALS:

Resignals if unable to handle error.

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION CTRLZ_COND_HANDLER (
                                   VAR SIGARGS : SIGARR;
                                   VAR MECHARGS : MECHARR
                                     ) : INTEGER;

    BEGIN

        { +
        If we're already unwinding, skip everything.
        - }
        IF NOT (
        (LIB$MATCH_COND (SIGARGS[1],SS$_UNWIND))
        ) THEN

        BEGIN

            { +
            Check for the ^Z "error".
            - }
            IF NOT (
            (LIB$MATCH_COND (SIGARGS[1],EDF$_CTRLZ))
            ) THEN

            BEGIN

                { +
                Tell the user what the disaster was.
                - }
                SIGARGS[0] := SIGARGS[0] - 2;
                $PUTMSG (SIGARGS);
                SIGARGS[0] := SIGARGS[0] + 2;

                { +
                Wait for the user to see what happened.
                - }
                LIB$WAIT (3.0);

            END;    { IF NOT LIB$MATCH_COND }

            { +
            Put the terminal straight.
            And close any files open to the terminal.
            - }
            IF NOT AUTO_TUNE THEN

            BEGIN

                EDF$RESET_SCROLL;

                IF DEST_IS_TERMINAL THEN

                    CLOSE (FDL_DEST,ERROR := CONTINUE);

            END;

            { +
            Unwind (pop up) to the caller of the handler establisher.
            - }
            $UNWIND;

            { +
            The function value is ignored if we did an unwind.
            - }
            CTRLZ_COND_HANDLER      := SS$_RESIGNAL;

        END;        { IF NOT UNWINDING }

    END;    { CTRLZ_COND_HANDLER }
```

```
{ ++

RMS_INPUT_COND_HANDLER -- Handle input file errors.

This routine checks for recoverable input errors from RMS files.

CALLING SEQUENCE:

LIB$SIGNAL;

INPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT INPUTS:

TAB
ANSI_REVERSE

OUTPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT OUTPUTS:

RMS_INPUT_ERROR
SYS$OUTPUT:, if the error is one we handle.

ROUTINES CALLED:

DELAY
LIB$MATCH_COND
SYS$UNWIND

ROUTINE VALUE:

SS$_RESIGNAL if unable to handle error.  N/A if able (ignored on unwind).

SIGNALS:

Resignals if unable to handle error.

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION RMS_INPUT_COND_HANDLER (
                              VAR SIGARGS : SIGARR;
                            VAR MECHARGS : MECHARR
                                  ) : INTEGER;

VAR
    FILENAME_PTR        : DESCRIPTOR_PTR;
    SEVERITY            : INTEGER;
    NEW_SEV             : INTEGER;

BEGIN

    { +
    If we're already unwinding, skip everything.
    - }
    IF NOT (
    (LIB$MATCH_COND (SIGARGS[1],SS$_UNWIND))
    ) THEN

    BEGIN

        RMS_INPUT_ERROR := TRUE;

        { +
        Find out the severity of the error.
        - }
        SEVERITY  := LIB$EXTZV (STS$V_SEVERITY,STS$S_SEVERITY,SIGARGS[1]);

        { +
        Show the user what's wrong, unless it'll come out on exit anyway.
        - }
        IF SEVERITY <> STS$K_SEVERE THEN

        BEGIN

            SIGARGS[0] := SIGARGS[0] - 2;
            $PUTMSG (SIGARGS);
            SIGARGS[0] := SIGARGS[0] + 2;

        END;

        { +
        Don't continue editing if this was a bad error.
        - }
        IF (SEVERITY IN [ STS$K_ERROR, STS$K_SEVERE ]) THEN

            EDITING     := FALSE;

        { +
        Unwind if it's a file-not-found (only for definition file).
        Otherwise, let EDF exit on bad errors.
        - }
        IF (
        ((SIGARGS[5] = RMS$_FNF) OR (SIGARGS[5] = SS$_NOSUCHFILE))
        AND
        (NOT ANALYSIS_ONLY)
        ) THEN
```

```
                    BEGIN
                    { +
                    Keep editing;
                    Make the FDL error informational;
                    Tell the user what file wasn't found;
                    Unwind (pop up) to the caller of the handler establisher.
                    - }
                    IF NOT (AUTO_TUNE)
                    THEN
                        BEGIN
                        EDITING := TRUE;
                        NEW_SEV := STS$K_INFO;
                        LIB$INSV (NEW_SEV,STS$V_SEVERITY,STS$S_SEVERITY,SIGARGS[1]);
                        CHFFLAGS          := 0;
                        WRITEV (OUT_LINE,CRLF);
                        LIB$PUT_LINE(OUT_LINE,ONE,CHFFLAGS);

                        FILENAME_PTR      := SIGARGS[3]::DESCRIPTOR_PTR;
                        WRITEV (OUT_LINE,CRLF_SHIFT,
                        FILENAME_PTR^.DSC$A_POINTER^:FILENAME_PTR^.DSC$W_LENGTH,
                        ' will be created.');
                        LIB$PUT_LINE(OUT_LINE,ONE,CHFFLAGS);
                        END
                    ELSE
                        BEGIN
                        SIGARGS[0] := SIGARGS[0] - 2;
                        $PUTMSG (SIGARGS);
                        SIGARGS[0] := SIGARGS[0] + 2;
                        END;
                    $UNWIND;
                END;  { if sigargs }
                { +
                The function value is ignored if we did an unwind.
                - }
                RMS_INPUT_COND_HANDLER  := SS$_CONTINUE;

            END;        { IF NOT UNWINDING }

        END;    { RMS_INPUT_COND_HANDLER }
```

```
{ ++

SYS$INPUT_COND_HANDLER -- Check for recoverable typing errors.

This routine handles Pascal input errors caused by user garbage.

CALLING SEQUENCE:

LIB$SIGNAL;

INPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT INPUTS:

CONTROL_Z
TAB
ANSI_REVERSE

OUTPUT PARAMETERS:

SIGARGS
MECHARGS

IMPLICIT OUTPUTS:

CONTROL_ZEE_TYPED
ERR_CHAR
QUESTION_TYPED
TEMP_FULL_PROMPT
SYS$INPUT_ERROR
SYS$OUTPUT:, if the error is one we can handle.

ROUTINES CALLED:

DELAY
LIB$MATCH_COND
SYS$UNWIND

ROUTINE VALUE:

SS$_RESIGNAL, if not unwinding.  N/A if it is unwinding.

SIGNALS:

Resignals if it can't process the signal.

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION SYS$INPUT_COND_HANDLER (
                            VAR SIGARGS : SIGARR;
                            VAR MECHARGS : MECHARR
                                ) : INTEGER;

VAR
    TEMP_UNSIGNED       : UNSIGNED;

BEGIN

    { +
    If we're already unwinding, skip everything.
    - }
    IF NOT (
    (LIB$MATCH_COND (SIGARGS[1],SS$_UNWIND))
    ) THEN

    BEGIN

        { +
        Check for bad typed input.
        - }
        IF (

        (LIB$MATCH_COND (SIGARGS[1],PAS$_GETAFTEOF))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_SUBASGVAL))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_AMBVALENU))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_INVSYNENU))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_INVSYNINT))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_INVSYNREA))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_INVSYNUNS))
        OR
        (LIB$MATCH_COND (SIGARGS[1],PAS$_NOTVALTYP))

        OR

        (LIB$MATCH_COND (SIGARGS[1],EDF$_CTRLZ))
        OR
        (LIB$MATCH_COND (SIGARGS[1],EDF$_AMBIG))
        OR
        (LIB$MATCH_COND (SIGARGS[1],EDF$_BADSYNTAX))
        OR
        (LIB$MATCH_COND (SIGARGS[1],EDF$_BADVALUE))
        OR
        (LIB$MATCH_COND (SIGARGS[1],EDF$_NODEFAULT))

        ) THEN

        BEGIN

            CONTROL_ZEE_TYPED   := LIB$MATCH_COND (SIGARGS[1],EDF$_CTRLZ);
```

```

{ +
Fudge for top-level ^Z exiting.
- }
IF MAIN_LEVEL THEN

    MAIN_CTRLZ                  := CONTROL_ZEE_TYPED;

{ +
If it was ^Z, don't look at the input string - there's nothing there.
- }
IF CONTROL_ZEE_TYPED THEN

    ERR_CHAR                    := CONTROL_Z

ELSE

    { +
    Get the offending character to see what it is.
    - }
    ERR_CHAR                    := INPUT_STRING[1];

{ +
One "garbage" character is "?" - which causes flags to get set.
- }
IF ERR_CHAR = QUESTION_MARK THEN

BEGIN

    QUESTION_TYPED              := TRUE;
    TEMP_FULL_PROMPT            := TRUE;

END

ELSE

    QUESTION_TYPED  := FALSE;

{ +
Tell the user he messed up, if he didn't type control/Z or "?".
- }
IF NOT ( CONTROL_ZEE_TYPED OR QUESTION_TYPED ) THEN

BEGIN

    { +
    Fetch the token that messed up.
    - }
    TEMP_DESCRIPTOR                     := NULL_STRING;
    TEMP_DESCRIPTOR.DSC$A_POINTER       := PARAM_BLOCK.TPA$L_TOKENPTR;
    TEMP_UNSIGNED                       := PARAM_BLOCK.TPA$L_TOKENCNT;
    TEMP_DESCRIPTOR.DSC$W_LENGTH        := TEMP_UNSIGNED::WORD;

    { +
    Print out the appropriate error message.
    - }
    IF (LIB$MATCH_COND (SIGARGS[1],EDF$_NODEFAULT)) THEN
```

```

            WRITEV (OUT_LINE,SHIFT,
            ' You must provide an answer here (or ^Z for Main Menu). ')

        ELSE IF (LIB$MATCH_COND (SIGARGS[1],EDF$_AMBIG)) THEN

            WRITEV (OUT_LINE,SHIFT,
            ' "',TEMP_DESCRIPTOR.DSC$A_POINTER^:
            TEMP_DESCRIPTOR.DSC$W_LENGTH,
            '"' is ambiguous in this context. ')

        ELSE IF (LIB$MATCH_COND (SIGARGS[1],EDF$_BADSYNTAX)) THEN

            WRITEV (OUT_LINE,SHIFT,
            ' "',TEMP_DESCRIPTOR.DSC$A_POINTER^:
            TEMP_DESCRIPTOR.DSC$W_LENGTH,
            '"' contains a syntax error. ')

        ELSE IF (LIB$MATCH_COND (SIGARGS[1],EDF$_BADVALUE)) THEN

            WRITEV (OUT_LINE,SHIFT,
            ' "',TEMP_DESCRIPTOR.DSC$A_POINTER^:
            TEMP_DESCRIPTOR.DSC$W_LENGTH,
            '"' is not appropriate in this context. ');

        CHFFLAGS        := SCR$M_REVERSE;
        LIB$PUT_LINE(OUT_LINE,ONE,CHFFLAGS);
        STR$FREE1_DX (INPUT_DESC);

        { +
        Let the user see the message.
        - }
        LIB$WAIT (2.0);

        { +
        Give the user some help.
        - }
        QUESTION_TYPED          := TRUE;
        TEMP_FULL_PROMPT        := TRUE;

    END;

    { +
    Flag the error and unwind back to the caller of the establisher.
    - }
    SYS$INPUT_ERROR     := TRUE;

    IF NOT CONTROL_ZEE_TYPED THEN

        { +
        Unwind (pop up) to the caller of the handler establisher.
        - }
        $UNWIND;

END;

{ +
        If we unwound, the function value will be ignored.
        If we didn't, we couldn't handle the error, so resignal.
        - }
        SYS$INPUT_COND_HANDLER  := SS$_RESIGNAL;

    END;       { IF NOT UNWINDING }

END;    { SYS$INPUT_COND_HANDLER }

END.
    { End of file: SRC$:EDFCHF.PAS }
```

```
                                                                              .TITLE   EDFCHF
                                                                              .IDENT   \V04-000\

                                                               00000          .PSECT   $CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2

    00000000   00000000   00000000   00000000   00000014       00000 C.AAA:   .LONG    ^X14,0,0,0,0,0,0,0
                          00000000   00000000   00000000       00014
  74  61  65  72  63  20  65  62  20  6C  6C  69  77  20       00020 C.AAB:   .ASCII   \ will be created.\<0><0><0>
                                  00  00  00  2E  64  65       0002E
  76  6F  72  70  20  74  73  75  6D  20  75  6F  59  20       00034 C.AAC:   .ASCII   \ You must provide an answer here (or ^Z \-
  20  72  65  77  73  6E  61  20  6E  61  20  65  64  69       00042                   \for Main Menu). \
  6F  66  20  5A  5E  20  72  6F  28  20  65  72  65  68       00050
  20  2E  29  75  6E  65  4D  20  6E  69  61  4D  20  72       0005E
                                          00  00  22  20       0006C C.AAD:   .ASCII   \."\<0><0>
  73  75  6F  75  67  69  62  6D  61  20  73  69  20  22       00070 C.AAE:   .ASCII   \" is ambiguous in this context. \
  65  74  6E  6F  63  20  73  69  68  74  20  6E  69  20       0007E
                                          20  2E  74  78       0008C
                                          00  00  22  20       00090 C.AAF:   .ASCII   \."\<0><0>
  73  20  61  20  73  6E  69  61  74  6E  6F  63  20  22       00094 C.AAG:   .ASCII   \" contains a syntax error. \<0>
  00  20  2E  72  6F  72  72  65  20  78  61  74  6E  79       000A2
                                          00  00  22  20       000B0 C.AAH:   .ASCII   \."\<0><0>
  6F  72  70  70  61  20  74  6F  6E  20  73  69  20  22       000B4 C.AAI:   .ASCII   \" is not appropriate in this context. \
  73  69  68  74  20  6E  69  20  65  74  61  69  72  70       000C2
                  20  2E  74  78  65  74  6E  6F  63  20       000D0

                                                               00000 CTRLZ_COND_HANDLER:                                       ; 0139
                                                          0000 00000          .WORD    ^M<>
                                            00000920  8F    DF 00002          PUSHAL   #2336                                   ; 0149
                                        50        04  AC    D0 00008          MOVL     4(R12),R0
                                                  04  A0    9F 0000C          PUSHAB   4(R0)
                             00000000G  EF            02    FB 0000F          CALLS    #2,LIB$MATCH_COND
                                        00V           50    E8 00016          BLBS     R0,11$
                                            00B3804B  8F    DF 00019          PUSHAL   #11763787                               ; 0158
                                        50        04  AC    D0 0001F          MOVL     4(R12),R0
                                                  04  A0    9F 00023          PUSHAB   4(R0)
                             00000000G  EF            02    FB 00026          CALLS    #2,LIB$MATCH_COND
                                        00V           50    E8 0002D          BLBS     R0,4$
                                    04  BC            02    C2 00030          SUBL2    #2,@4(R12)                              ; 0167
                                                      00    DD 00034          PUSHL    #0                                      ; 0168
                                                      00    DD 00036          PUSHL    #0
                                                      00    DD 00038          PUSHL    #0
                                                  04  AC    DD 0003A          PUSHL    4(R12)
                             00000000G  EF            04    FB 0003D          CALLS    #4,SYS$PUTMSG
                                    04  BC            02    C0 00044          ADDL2    #2,@4(R12)                              ; 0169
                                            00004140  8F    DF 00048          PUSHAF   #^F3.0                                  ; 0174
                             00000000G  EF            01    FB 0004E          CALLS    #1,LIB$WAIT
                          00V00000000G  EF            00    E0 00055 4$:      BBS      #0,AUTO_TUNE,9$                         ; 0182
                             00000000G  EF            00    FB 0005D          CALLS    #0,EDF$RESET_SCROLL                     ; 0186
                          00V00000000G  EF            00    E1 00064          BBC      #0,DEST_IS_TERMINAL,9$                  ; 0188
                                                 00V  AF    9F 0006C          PUSHAB   9$                                      ; 0190
                                                      19    DD 0006F          PUSHL    #25
                                           00000000G  EF    9F 00071          PUSHAB   FDL_DEST
                             00000000G  EF            03    FB 00077          CALLS    #3,PAS$CLOSE2
                                                      00    DD 0007E 9$:      PUSHL    #0                                      ; 0197
                                                      00    DD 00080          PUSHL    #0
                             00000000G  EF            02    FB 00082          CALLS    #2,SYS$UNWIND
                                        50      0918  8F    3C 00089          MOVZWL   #2328,CTRLZ_COND_HANDLER                ; 0202
```

```
                                      04 0008E 11$:    RET                                  ; 0206

; Routine Size: 143 bytes,    Routine Base: $CODE + 000DA

                                         00000 RMS_INPUT_COND_HANDLER:                      ; 0258
                                    0004 00000         .WORD    ^M<R2>
              5E                 08   C2 00002         SUBL2    #8,SP
                    00000920     8F   DF 00005         PUSHAL   #2336                       ; 0273
              50          04     AC   D0 0000B         MOVL     4(R12),R0
                          04     A0   9F 0000F         PUSHAB   4(R0)
   00000000G  EF                 02   FB 00012         CALLS    #2,LIB$MATCH_COND
              03                 50   E9 00019         BLBC     R0,.+3
                              0000V   31 0001C         BRW      18$
   00000000G  EF                 01   90 0001F         MOVB     #1,RMS_INPUT_ERROR          ; 0279
              50          04     AC   D0 00026         MOVL     4(R12),R0                   ; 0284
                          04     A0   9F 0002A         PUSHAB   4(R0)
                    00000003     8F   DF 0002D         PUSHAL   #3
                    00000000     8F   DF 00033         PUSHAL   #0
   00000000G  EF                 03   FB 00039         CALLS    #3,LIB$EXTZV
              52                 50   D0 00040         MOVL     R0,SEVERITY
              04                 52   D1 00043         CMPL     SEVERITY,#4                 ; 0289
                                00V   13 00046         BEQL     4$
          04  BC                 02   C2 00048         SUBL2    #2,@4(R12)                  ; 0293
                                 00   DD 0004C         PUSHL    #0                          ; 0294
                                 00   DD 0004E         PUSHL    #0
                                 00   DD 00050         PUSHL    #0
                          04     AC   DD 00052         PUSHL    4(R12)
   00000000G  EF                 04   FB 00055         CALLS    #4,SYS$PUTMSG
          04  BC                 02   C0 0005C         ADDL2    #2,@4(R12)                  ; 0295
    00000100  8F                 52   D1 00060 4$:     CMPL     SEVERITY,#256               ; 0302
                                00V   1E 00067         BGEQU    6$
 00VFFFFFE26  EF                 52   E1 00069         BBC      SEVERITY,C.AAA,6$
                   00000000G     EF   94 00071         CLRB     EDITING                     ; 0304
              52          04     AC   D0 00077 6$:     MOVL     4(R12),R2                   ; 0310
    00018292  8F          14     A2   D1 0007B         CMPL     20(R2),#98962
                                00V   13 00083         BEQL     8$
              52          04     AC   D0 00085         MOVL     4(R12),R2
    00000910  8F          14     A2   D1 00089         CMPL     20(R2),#2320
                                 03   13 00091         BEQL     .+3
                              0000V   31 00093         BRW      17$
03 00000000G  EF                 00   E1 00096 8$:     BBC      #0,ANALYSIS_ONLY,.+3
                              0000V   31 0009E         BRW      17$
03 00000000G  EF                 00   E1 000A1         BBC      #0,AUTO_TUNE,.+3            ; 0322
                              0000V   31 000A9         BRW      13$
   00000000G  EF                 01   90 000AC         MOVB     #1,EDITING                  ; 0325
              52                 03   D0 000B3         MOVL     #3,NEW_SEV                  ; 0326
7E        04  AC                 04   C1 000B6         ADDL3    #4,4(R12),-(SP)             ; 0327
                    00000003     8F   DF 000BB         PUSHAL   #3
                    00000000     8F   DF 000C1         PUSHAL   #0
          FC  AD                 52   D0 000C7         MOVL     NEW_SEV,-4(FP)
                          FC     AD   9F 000CB         PUSHAB   -4(FP)
   00000000G  EF                 04   FB 000CE         CALLS    #4,LIB$INSV
                   00000000G     EF   D4 000D5         CLRL     CHFFLAGS                    ; 0328
                   00000000G     EF   B4 000DB         CLRW     OUT_LINE                    ; 0329
                   00000000G     EF   9F 000E1         PUSHAB   CRLF
                                 02   DD 000E7         PUSHL    #2
                   00000000G     EF   9F 000E9         PUSHAB   OUT_LINE
```

```
                000000FF  8F  DD 000EF       PUSHL   #255
00000000G  EF             04  FB 000F5       CALLS   #4,PAS$WRITEV_STRING
               00000000G  EF  9F 000FC       PUSHAB  CHFFLAGS                                  ; 0330
               00000000G  EF  9F 00102       PUSHAB  ONE
       F8  AD  0B2500FF   8F  D0 00108       MOVL    #186974463,-8(FP)
       FC  AD  00000000G  EF  9E 00110       MOVAB   OUT_LINE,-4(FP)
                      F8  AD  9F 00118       PUSHAB  -8(FP)
00000000G  EF             03  FB 0011B       CALLS   #3,LIB$PUT_LINE
       50             04  AC  D0 00122       MOVL    4(R12),R0                                 ; 0332
       52             0C  A0  D0 00126       MOVL    12(R0),FILENAME_PTR
               00000000G  EF  B4 0012A       CLRW    OUT_LINE                                  ; 0333
               00000000G  EF  9F 00130       PUSHAB  CRLF_SHIFT
                          06  DD 00136       PUSHL   #6
               00000000G  EF  9F 00138       PUSHAB  OUT_LINE
                000000FF  8F  DD 0013E       PUSHL   #255
00000000G  EF             04  FB 00144       CALLS   #4,PAS$WRITEV_STRING
       7E                 62  3C 0014B       MOVZWL  (FILENAME_PTR),-(SP)
                          00  DD 0014E       PUSHL   #0
                      04  B2  9F 00150       PUSHAB  @4(FILENAME_PTR)
                000000FF  8F  DD 00153       PUSHL   #255
               00000000G  EF  9F 00159       PUSHAB  OUT_LINE
                000000FF  8F  DD 0015F       PUSHL   #255
00000000G  EF             06  FB 00165       CALLS   #6,PAS$WRITEV_STRING
                FFFFFD45  EF  9F 0016C       PUSHAB  C.AAB
                          11  DD 00172       PUSHL   #17
               00000000G  EF  9F 00174       PUSHAB  OUT_LINE
                000000FF  8F  DD 0017A       PUSHL   #255
00000000G  EF             04  FB 00180       CALLS   #4,PAS$WRITEV_STRING
               00000000G  EF  9F 00187       PUSHAB  CHFFLAGS                                  ; 0336
               00000000G  EF  9F 0018D       PUSHAB  ONE
       F8  AD  0B2500FF   8F  D0 00193       MOVL    #186974463,-8(FP)
       FC  AD  00000000G  EF  9E 0019B       MOVAB   OUT_LINE,-4(FP)
                      F8  AD  9F 001A3       PUSHAB  -8(FP)
00000000G  EF             03  FB 001A6       CALLS   #3,LIB$PUT_LINE
                          00V 11 001AD       BRB     15$
       04  BC             02  C2 001AF 13$:  SUBL2   #2,@4(R12)                                ; 0340
                          00  DD 001B3       PUSHL   #0                                        ; 0341
                          00  DD 001B5       PUSHL   #0
                          00  DD 001B7       PUSHL   #0
                      04  AC  DD 001B9       PUSHL   4(R12)
00000000G  EF             04  FB 001BC       CALLS   #4,SYS$PUTMSG
       04  BC             02  C0 001C3       ADDL2   #2,@4(R12)                                ; 0342
                          00  DD 001C7 15$:  PUSHL   #0                                        ; 0344
                          00  DD 001C9       PUSHL   #0
00000000G  EF             02  FB 001CB       CALLS   #2,SYS$UNWIND
       50                 01  D0 001D2 17$:  MOVL    #1,RMS_INPUT_COND_HANDLER                 ; 0349
                              04 001D5 18$:  RET                                               ; 0353

; Routine Size: 470 bytes,    Routine Base: $CODE + 00169

                                 00000 SYS$INPUT_COND_HANDLER:                                 ; 0410
                            0FFC 00000       .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
       5E                 08  C2 00002       SUBL2   #8,SP
                00000920  8F  DF 00005       PUSHAL  #2336                                     ; 0423
       50             04  AC  D0 0000B       MOVL    4(R12),R0
                      04  A0  9F 0000F       PUSHAB  4(R0)
00000000G  EF             02  FB 00012       CALLS   #2,LIB$MATCH_COND
```

```
           03                50 E9 00019     BLBC    R0,.+3
                          0000V 31 0001C     BRW     30$
                00B38040     8F DF 0001F     PUSHAL  #11763776                               ; 0432
           50         04     AC D0 00025     MOVL    4(R12),R0
                      04     A0 9F 00029     PUSHAB  4(R0)
00000000G  EF                02 FB 0002C     CALLS   #2,LIB$MATCH_COND
       FC  AD                50 90 00033     MOVB    R0,-4(FP)
                00B38038     8F DF 00037     PUSHAL  #11763768
           50         04     AC D0 0003D     MOVL    4(R12),R0
                      04     A0 9F 00041     PUSHAB  4(R0)
00000000G  EF                02 FB 00044     CALLS   #2,LIB$MATCH_COND
       F8  AD                50 90 0004B     MOVB    R0,-8(FP)
                00B38030     8F DF 0004F     PUSHAL  #11763760
           50         04     AC D0 00055     MOVL    4(R12),R0
                      04     A0 9F 00059     PUSHAB  4(R0)
00000000G  EF                02 FB 0005C     CALLS   #2,LIB$MATCH_COND
           54                50 90 00063     MOVB    R0,R4
                00B38028     8F DF 00066     PUSHAL  #11763752
           50         04     AC D0 0006C     MOVL    4(R12),R0
                      04     A0 9F 00070     PUSHAB  4(R0)
00000000G  EF                02 FB 00073     CALLS   #2,LIB$MATCH_COND
           55                50 90 0007A     MOVB    R0,R5
                00B3804B     8F DF 0007D     PUSHAL  #11763787
           50         04     AC D0 00083     MOVL    4(R12),R0
                      04     A0 9F 00087     PUSHAB  4(R0)
00000000G  EF                02 FB 0008A     CALLS   #2,LIB$MATCH_COND
           56                50 90 00091     MOVB    R0,R6
                0021875C     8F DF 00094     PUSHAL  #2197340
           50         04     AC D0 0009A     MOVL    4(R12),R0
                      04     A0 9F 0009E     PUSHAB  4(R0)
00000000G  EF                02 FB 000A1     CALLS   #2,LIB$MATCH_COND
           57                50 90 000A8     MOVB    R0,R7
                00218754     8F DF 000AB     PUSHAL  #2197332
           50         04     AC D0 000B1     MOVL    4(R12),R0
                      04     A0 9F 000B5     PUSHAB  4(R0)
00000000G  EF                02 FB 000B8     CALLS   #2,LIB$MATCH_COND
           58                50 90 000BF     MOVB    R0,R8
                0021874C     8F DF 000C2     PUSHAL  #2197324
           50         04     AC D0 000C8     MOVL    4(R12),R0
                      04     A0 9F 000CC     PUSHAB  4(R0)
00000000G  EF                02 FB 000CF     CALLS   #2,LIB$MATCH_COND
           59                50 90 000D6     MOVB    R0,R9
                00218744     8F DF 000D9     PUSHAL  #2197316
           50         04     AC D0 000DF     MOVL    4(R12),R0
                      04     A0 9F 000E3     PUSHAB  4(R0)
00000000G  EF                02 FB 000E6     CALLS   #2,LIB$MATCH_COND
           5A                50 90 000ED     MOVB    R0,R10
                0021873C     8F DF 000F0     PUSHAL  #2197308
           50         04     AC D0 000F6     MOVL    4(R12),R0
                      04     A0 9F 000FA     PUSHAB  4(R0)
00000000G  EF                02 FB 000FD     CALLS   #2,LIB$MATCH_COND
           5B                50 90 00104     MOVB    R0,R11
                00218734     8F DF 00107     PUSHAL  #2197300
           50         04     AC D0 0010D     MOVL    4(R12),R0
                      04     A0 9F 00111     PUSHAB  4(R0)
00000000G  EF                02 FB 00114     CALLS   #2,LIB$MATCH_COND
           52                50 90 0011B     MOVB    R0,R2
```

```
                  0021B9F4    8F DF 0011E        PUSHAL  #219B004
              50        04    AC D0 00124        MOVL    4(R12),R0
                        04    A0 9F 00128        PUSHAB  4(R0)
    00000000G EF              02 FB 0012B        CALLS   #2,LIB$MATCH_COND
              53              50 90 00132        MOVB    R0,R3
                  0021BEDC    8F DF 00135        PUSHAL  #2211548
              50        04    AC D0 0013B        MOVL    4(R12),R0
                        04    A0 9F 0013F        PUSHAB  4(R0)
    00000000G EF              02 FB 00142        CALLS   #2,LIB$MATCH_COND
              50              53 88 00149        BISB2   R3,R0
              50              52 88 0014C        BISB2   R2,R0
              50              5B 88 0014F        BISB2   R11,R0
              50              5A 88 00152        BISB2   R10,R0
              50              59 88 00155        BISB2   R9,R0
              50              58 88 00158        BISB2   R8,R0
              50              57 88 0015B        BISB2   R7,R0
              50              56 88 0015E        BISB2   R6,R0
              50              55 88 00161        BISB2   R5,R0
              50              54 88 00164        BISB2   R4,R0
              50        F8    AD 88 00167        BISB2   -8(FP),R0
              50        FC    AD 88 0016B        BISB2   -4(FP),R0
              03              50 E8 0016F        BLBS    R0,.+3
                           0000V 31 00172        BRW     29$
                  00B3804B    8F DF 00175        PUSHAL  #11763787                                    ; 0466
              50        04    AC D0 0017B        MOVL    4(R12),R0
                        04    A0 9F 0017F        PUSHAB  4(R0)
    00000000G EF              02 FB 00182        CALLS   #2,LIB$MATCH_COND
    00000000G EF              50 90 00189        MOVB    R0,CONTROL_ZEE_TYPED
00v000000000G EF              00 E1 00190        BBC     #0,MAIN_LEVEL,4$                             ; 0471
    00000000G EF 00000000G    EF 90 00198        MOVB    CONTROL_ZEE_TYPED,MAIN_CTRLZ                 ; 0473
00v000000000G EF              00 E1 001A3 4$:    BBC     #0,CONTROL_ZEE_TYPED,6$                      ; 0478
    00000000G EF 00000000G    EF 90 001AB        MOVB    CONTROL_Z,ERR_CHAR                           ; 0480
                             00V 11 001B6        BRB     7$
    00000000G EF 00000000G    EF 90 001B8 6$:    MOVB    INPUT_STRING,ERR_CHAR                        ; 0487
    00000000G EF 00000000G    EF 91 001C3 7$:    CMPB    ERR_CHAR,QUESTION_MARK                       ; 0492
                             00V 12 001CE        BNEQ    9$
    00000000G EF              01 90 001D0        MOVB    #1,QUESTION_TYPED                            ; 0496
    00000000G EF              01 90 001D7        MOVB    #1,TEMP_FULL_PROMPT                          ; 0497
                             00V 11 001DE        BRB     10$
                 00000000G    EF 94 001E0 9$:    CLRB    QUESTION_TYPED                               ; 0503
 03 00000000G EF              00 E1 001E6 10$:   BBC     #0,CONTROL_ZEE_TYPED,..+3                    ; 0508
                           0000V 31 001EE        BRW     25$
 03 00000000G EF              00 E1 001F1        BBC     #0,QUESTION_TYPED,..+3
                           0000V 31 001F9        BRW     25$
    00000000G EF 00000000G    EF 7D 001FC        MOVQ    NULL_STRING,TEMP_DESCRIPTOR                  ; 0515
    00000004G EF 00000014G    EF D0 00207        MOVL    PARAM_BLOCK+20,TEMP_DESCRIPTOR+4             ; 0516
              50 00000010G    EF D0 00212        MOVL    PARAM_BLOCK+16,TEMP_UNSIGNED                 ; 0517
    00000000G EF              50 B0 00219        MOVW    TEMP_UNSIGNED,TEMP_DESCRIPTOR                ; 0518
                  00B38040    8F DF 00220        PUSHAL  #11763776                                    ; 0523
              50        04    AC D0 00226        MOVL    4(R12),R0
                        04    A0 9F 0022A        PUSHAB  4(R0)
    00000000G EF              02 FB 0022D        CALLS   #2,LIB$MATCH_COND
              00V             50 E9 00234        BLBC    R0,14$
                 00000000G    EF B4 00237        CLRW    OUT_LINE                                     ; 0525
                 00000000G    EF 9F 0023D        PUSHAB  SHIFT
                              04 DD 00243        PUSHL   #4
                 00000000G    EF 9F 00245        PUSHAB  OUT_LINE
```

```
                        000000FF    8F   DD 0024B          PUSHL    #255
          00000000G  EF             04   FB 00251          CALLS    #4,PAS$WRITEV_STRING
                        FFFFFA97    EF   9F 00258          PUSHAB   C.AAC
                                    38   DD 0025E          PUSHL    #56
                        00000000G   EF   9F 00260          PUSHAB   OUT_LINE
                        000000FF    8F   DD 00266          PUSHL    #255
          00000000G  EF             04   FB 0026C          CALLS    #4,PAS$WRITEV_STRING
                                0000V    31 00273          BRW      23$
                        00B38028    8F   DF 00276 14$:     PUSHAL   #11763752                                    ; 0528
                     50       04    AC   D0 0027C          MOVL     4(R12),R0
                              04    A0   9F 00280          PUSHAB   4(R0)
          00000000G  EF             02   FB 00283          CALLS    #2,LIB$MATCH_COND
                     03             50   E8 0028A          BLBS     R0,.+3
                                0000V    31 0028D          BRW      16$
                        00000000G   EF   B4 00290          CLRW     OUT_LINE                                     ; 0530
                        00000000G   EF   9F 00296          PUSHAB   SHIFT
                                    04   DD 0029C          PUSHL    #4
                        00000000G   EF   9F 0029E          PUSHAB   OUT_LINE
                        000000FF    8F   DD 002A4          PUSHL    #255
          00000000G  EF             04   FB 002AA          CALLS    #4,PAS$WRITEV_STRING
                        FFFFFA76    EF   9F 002B1          PUSHAB   C.AAD
                                    02   DD 002B7          PUSHL    #2
                        00000000G   EF   9F 002B9          PUSHAB   OUT_LINE
                        000000FF    8F   DD 002BF          PUSHL    #255
          00000000G  EF             04   FB 002C5          CALLS    #4,PAS$WRITEV_STRING
                     7E 00000000G   EF   3C 002CC          MOVZWL   TEMP_DESCRIPTOR,-(SP)
                                    00   DD 002D3          PUSHL    #0
                     50 00000004G   EF   D0 002D5          MOVL     TEMP_DESCRIPTOR+4,R0
                                    60   9F 002DC          PUSHAB   (R0)
                        000000FF    8F   DD 002DE          PUSHL    #255
                        00000000G   EF   9F 002E4          PUSHAB   OUT_LINE
                        000000FF    8F   DD 002EA          PUSHL    #255
          00000000G  EF             06   FB 002F0          CALLS    #6,PAS$WRITEV_STRING
                        FFFFFA34    EF   9F 002F7          PUSHAB   C.AAE
                                    20   DD 002FD          PUSHL    #32
                        00000000G   EF   9F 002FF          PUSHAB   OUT_LINE
                        000000FF    8F   DD 00305          PUSHL    #255
          00000000G  EF             04   FB 0030B          CALLS    #4,PAS$WRITEV_STRING
                                0000V    31 00312          BRW      23$
                        00B38030    8F   DF 00315 16$:     PUSHAL   #11763760                                    ; 0535
                     50       04    AC   D0 0031B          MOVL     4(R12),R0
                              04    A0   9F 0031F          PUSHAB   4(R0)
          00000000G  EF             02   FB 00322          CALLS    #2,LIB$MATCH_COND
                     03             50   E8 00329          BLBS     R0,.+3
                                0000V    31 0032C          BRW      18$
                        00000000G   EF   B4 0032F          CLRW     OUT_LINE                                     ; 0537
                        00000000G   EF   9F 00335          PUSHAB   SHIFT
                                    04   DD 0033B          PUSHL    #4
                        00000000G   EF   9F 0033D          PUSHAB   OUT_LINE
                        000000FF    8F   DD 00343          PUSHL    #255
          00000000G  EF             04   FB 00349          CALLS    #4,PAS$WRITEV_STRING
                        FFFFF9FB    EF   9F 00350          PUSHAB   C.AAF
                                    02   DD 00356          PUSHL    #2
                        00000000G   EF   9F 00358          PUSHAB   OUT_LINE
                        000000FF    8F   DD 0035E          PUSHL    #255
          00000000G  EF             04   FB 00364          CALLS    #4,PAS$WRITEV_STRING
                     7E 00000000G   EF   3C 0036B          MOVZWL   TEMP_DESCRIPTOR,-(SP)
```

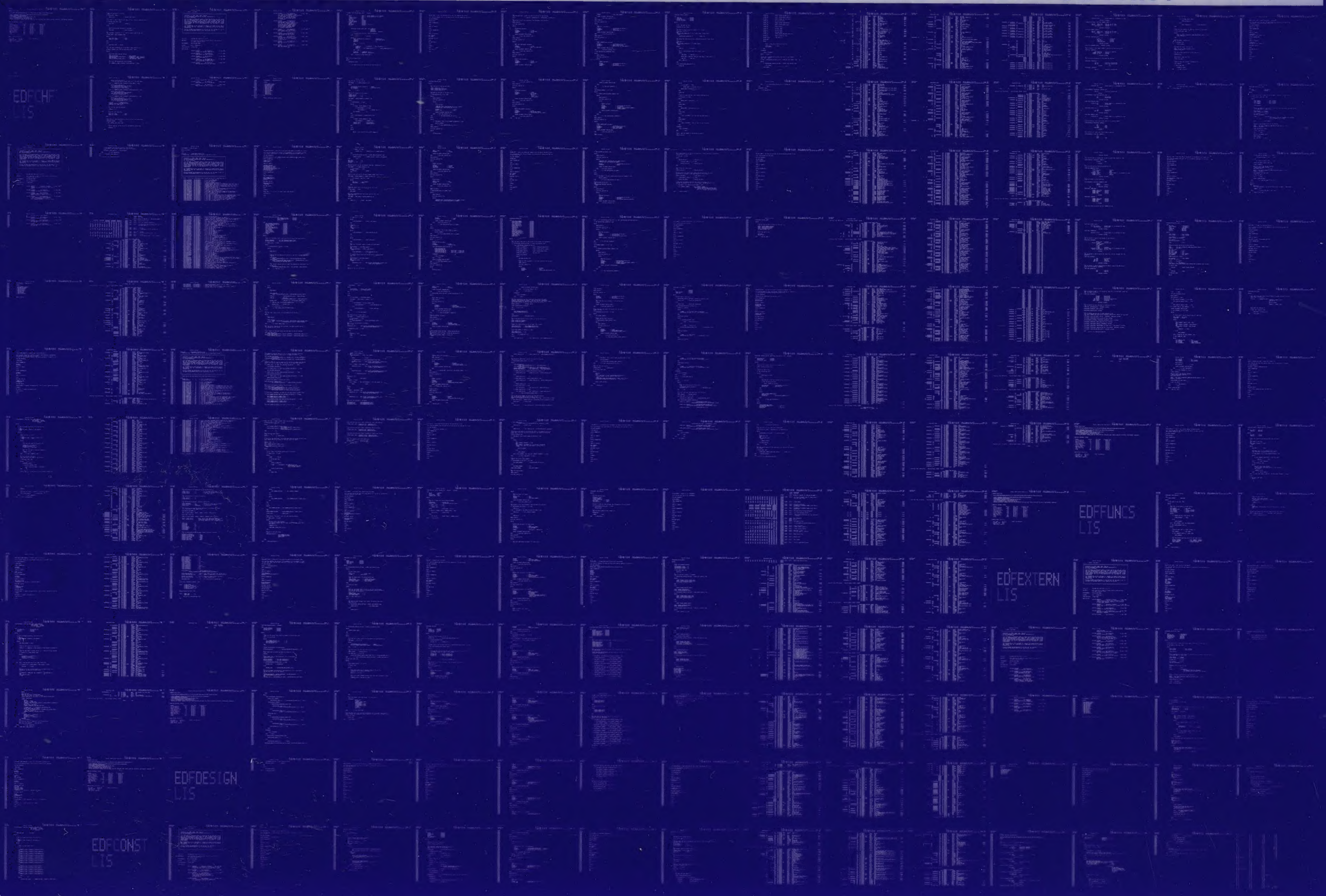
```
                          00    DD 00372          PUSHL    #0
           50 00000004G   EF    D0 00374          MOVL     TEMP_DESCRIPTOR+4,R0
                          60    9F 0037B          PUSHAB   (R0)
              000000FF    8F    DD 0037D          PUSHL    #255
              00000000G   EF    9F 00383          PUSHAB   OUT_LINE
              000000FF    8F    DD 00389          PUSHL    #255
00000000G  EF             06    FB 0038F          CALLS    #6,PAS$WRITEV_STRING
              FFFFF9B9    EF    9F 00396          PUSHAB   C.AAG
                          1B    DD 0039C          PUSHL    #27
              00000000G   EF    9F 0039E          PUSHAB   OUT_LINE
              000000FF    8F    DD 003A4          PUSHL    #255
00000000G  EF             04    FB 003AA          CALLS    #4,PAS$WRITEV_STRING
                      0000V     31 003B1          BRW      23$
              00B38038    8F    DF 003B4 18$:     PUSHAL   #11763768                      ; 0542
           5C        04   AC    D0 003BA          MOVL     4(R12),R12
                     04   AC    9F 003BE          PUSHAB   4(R12)
00000000G  EF             02    FB 003C1          CALLS    #2,LIB$MATCH_COND
           03             50    E8 003C8          BLBS     R0,.+3
                      0000V     31 003CB          BRW      23$
              00000000G   EF    B4 003CE          CLRW     OUT_LINE                       ; 0544
              00000000G   EF    9F 003D4          PUSHAB   SHIFT
                          04    DD 003DA          PUSHL    #4
              00000000G   EF    9F 003DC          PUSHAB   OUT_LINE
              000000FF    8F    DD 003E2          PUSHL    #255
00000000G  EF             04    FB 003E8          CALLS    #4,PAS$WRITEV_STRING
              FFFFF97C    EF    9F 003EF          PUSHAB   C.AAH
                          02    DD 003F5          PUSHL    #2
              00000000G   EF    9F 003F7          PUSHAB   OUT_LINE
              000000FF    8F    DD 003FD          PUSHL    #255
00000000G  EF             04    FB 00403          CALLS    #4,PAS$WRITEV_STRING
           7E 00000000G   EF    3C 0040A          MOVZWL   TEMP_DESCRIPTOR,-(SP)
                          00    DD 00411          PUSHL    #0
           50 00000004G   EF    D0 00413          MOVL     TEMP_DESCRIPTOR+4,R0
                          60    9F 0041A          PUSHAB   (R0)
              000000FF    8F    DD 0041C          PUSHL    #255
              00000000G   EF    9F 00422          PUSHAB   OUT_LINE
              000000FF    8F    DD 00428          PUSHL    #255
00000000G  EF             06    FB 0042E          CALLS    #6,PAS$WRITEV_STRING
              FFFFF93A    EF    9F 00435          PUSHAB   C.AAI
                          26    DD 0043B          PUSHL    #38
              00000000G   EF    9F 0043D          PUSHAB   OUT_LINE
              000000FF    8F    DD 00443          PUSHL    #255
00000000G  EF             04    FB 00449          CALLS    #4,PAS$WRITEV_STRING
00000000G  EF             02    D0 00450 23$:     MOVL     #2,CHFFLAGS                    ; 0549
              00000000G   EF    9F 00457          PUSHAB   CHFFLAGS                       ; 0550
              00000000G   EF    9F 0045D          PUSHAB   ONE
       F8  AD 0B2500FF    8F    D0 00463          MOVL     #186974463,-8(FP)
       FC  AD 00000000G   EF    9E 0046B          MOVAB    OUT_LINE,-4(FP)
                     F8   AD    9F 00473          PUSHAB   -8(FP)
00000000G  EF             03    FB 00476          CALLS    #3,LIB$PUT_LINE
              00000000G   EF    9F 0047D          PUSHAB   INPUT_DESC                     ; 0551
00000000G  EF             01    FB 00483          CALLS    #1,STR$FREE1_DX
              00004100    8F    DF 0048A          PUSHAF   #^F2.0                         ; 0556
00000000G  EF             01    FB 00490          CALLS    #1,LIB$WAIT
00000000G  EF             01    90 00497          MOVB     #1,QUESTION_TYPED              ; 0561
00000000G  EF             01    90 0049E          MOVB     #1,TEMP_FULL_PROMPT            ; 0562
00000000G  EF             01    90 004A5 25$:     MOVB     #1,SYS$INPUT_ERROR             ; 0569
```

```
                   00V00000000G  EF            00   E0 004AC         BBS     #0,CONTROL_ZEE_TYPED,29$                 ; 0571
                                               00   DD 004B4         PUSHL   #0                                        ; 0576
                                               00   DD 004B6         PUSHL   #0
                     00000000G  EF             02   FB 004B8         CALLS   #2,SYS$UNWIND
                                50      0918   8F   3C 004BF 29$:    MOVZWL  #2328,SYS$INPUT_COND_HANDLER              ; 0584
                                                    04 004C4 30$:    RET                                               ; 0588

; Routine Size: 1221 bytes,    Routine Base: $CODE + 0033F

                                                       00804         .END
```

```
COMMAND QUALIFIERS

  PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS$:EDFCHF/OBJ=OBJ$:EDFCHF MSRC$:EDFCHF

  /CHECK=(NOBOUNDS,NOCASE_SELECTORS,NOOVERFLOW,NOPOINTERS,NOSUBRANGE)
  /DEBUG=(NOSYMBOLS,NOTRACEBACK)
  /ENVIRONMENT=_$255$DUA28:[EDF.OBJ]EDFCHF.PEN;1
  /LIST=_$255$DUA28:[EDF.LIS]EDFCHF.LIS;1
  /OBJECT=_$255$DUA28:[EDF.OBJ]EDFCHF.OBJ;1
  /NOCROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /MACHINE_CODE /NOOLD_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS


COMPILER INTERNAL TIMING

  Phase                   Faults       CPU Time       Elapsed Time
  Initialization              71       00:00.4           00:02.5
  Source Analysis            617       00:12.8           02:37.5
  Source Listing              40       00:00.9           00:02.2
  Tree Construction           76       00:00.5           00:01.0
  Flow Analysis                8       00:00.1           00:00.2
  Profit Analysis             14       00:00.2           00:00.2
  Context Analysis           614       00:04.5           00:07.4
  Name Packing                 2       00:00.1           00:00.1
  Code Selection              21       00:00.6           00:01.8
  Final                      145       00:02.3           00:08.3
  TOTAL                     1612       00:22.2           03:01.2


COMPILATION STATISTICS

  CPU Time:        00:22.2          (1596 Lines/Minute)
  Elapsed Time:    03:01.2
  Page Faults:     1612
  Compilation Complete
```

0126 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



EDFCHF
LIS

EDFFUNCS
LIS

EDFEXTERN
LIS

EDFDESIGN
LIS

EDFCONST
LIS